Bowsense – A minimalistic Approach to Wireless Motion Sensing

Hans WILMERS
Norwegian Centre for Technology
in Music and the Arts
(NOTAM)
Nedre Gate 5,
0551 Oslo – Norway
hans.wilmers@notam02.no

Abstract

A novel wireless motion sensing device for control of realtime electronics is presented.

The device consists of a small ($20 \times 40\,mm$) circuit board including all circuitry and battery, and contains a complete inertial measuring unit (IMU), measuring both acceleration and angular velocity in 3 dimensions. Data is transmitted wirelessly via a bluetooth link and can be read from applications either using bluetooth serial emulation, or using a small standalone server application that converts incoming data and sends reformatted data out via OSC.

Keywords

Wireless, Bluetooth, Bow, Motion, Controller

1 Introduction

In [1] a new sensorbow project was presented - a project that was started at the Norwegian Academy of Music in colaboration with NOTAM. The project is primarily focused on the artistic value of augmenting string instruments, but some of our findings can be applied to other controlling tasks, and the sensor device itself is versatile and flexible in design. In fact, the hardware and software design itself is open, and can be retrieved from the author's website.

In this article, the main focus is on the electronics used in the project, the technical challenges encountered on the way, and the solutions to them.

2 Sensing the Bow

Wireless motion sensing devices have existed for quite a while. Many of these have been used with bowed string instruments [2; 3; 4; 5], and several have had the aim of augmenting existing instruments by adding new controlling possibilities [3; 4; 5].

For our project, a new device was developed that could be adapted closely to the task. It needed to be small and inobtrusive, and it should be usable in real stage performance situations. In addition, the device should be easy to extend with external sensors such as pushbuttons and a finger pressure sensor.

A first model shown in [1] contains a 3D accelerometer, and a 2D gyroscope providing angular velocity. The second model now contains a 3rd gyroscope, enhancing the device to a complete IMU with 6 degrees of freedom. It is thus possible to calculate a good estimate for the orientation of the device in respect to the gravity vector.

3 Electronics

The first model contained a $1.5\,V$ AAA battery and a step-up converter. This solution was replaced by a Li-Polymer battery, reducing both weight and complexity of the design. A battery charger was integrated into the device, and the battery can be mounted in a way that enables "emergency exchange" in stage situations. The device can be charged from a standard mini-USB cable.

To reduce noise in the analog circuitry, the device contains a separate $3\,V$ low dropout regulator for the analog supply, and a $3.3\,V$ regulator for digital components, including the bluetooth module. With this circuitry, the device is able to run very near the cutoff-voltage of a typical Li-Polymer battery, and the resulting runtime is well over $2\,h$.

Function	Component
Microcontroller	C8051F530 / Silabs
Bluetooth Module	RN41 / Roving Networks
Accelerometer	ADXL330 / Analog Dev.
Gyroscope, X/Y	IDG300 / Invensense
Gyroscope, Z	LISY300AL / ST
Battery	LPP402025 / Varta
Battery Charger	MCP73831 / Microchip

Table 1: Components

The device is built around a C8051F530 microcontroller from Silabs, which contains a 25MIPS core and a 12 bit A/D-converter that can be used with any of the unused port pins. With a total of 16 port pins, and 5 of these used for digital signals, this leaves up to 11 inputs for analog values. Of these, one is used to sample the battery voltage, 3 for the 3D accelerometer, and another 3 for the gyroscopes. The remaining 4 unused inputs are made available at the edge of the PCB, and can be used to connect additional sensors to the device.

As a bluetooth transmitter, the RN41 module made by Roving Networks is used. The RN41 has a max. range of 100 m in open air. Data is sent into the module by a serial port with 115kbaud. As the bluetooth data rate drops considerably with increasing distance, hardware flow control is used to avoid data loss.

In the choice of sensors, both technical specifications and resulting simplicity of the design were important criteria. All sensors in this design can run from 3V, so a single analog supply was sufficient. The sensor outputs are connected to analog inputs of the microcontroller, with only a few additional passive components.

As accelerometer, an ADXL330 from Analog Devices is used. It measures up to $\pm 3\,g$ ([1]) along 3 axes X, Y (along the PCB surface) and Z (perpendicular to the surface). To measure angular velocity in 3 dimensions, two gyroscopes are used: An IDG300 from Invensense, measuring up to $\pm 500\,deg/s$ around the X and Y axes, and a LISY300AL from ST, measuring up to $\pm 300\,deg/s$ around the Z axis. The battery used (Varta LPP402025) is a single cell Li-Polymer battery with a capacity of 150 mAh, and a weight of under 4 g ([2]). The total weight of the sensor board inkl. battery is approximately 12 g.

The electronics layout was done with Eagle 5.3 professional, but as only two PCB layers are used, further work can be done with the free edition Eagle Light. Special consideration has been made to make handsoldering of the board possible, the necessary experience and tools provided.

4 Transfer Protocol

Data is sent via bluetooth, using the Serial Port Profile (SPP). A very simple binary packet protocol is used, which was derived from the protocol used by Dan Overholt on the CUI [6] device. Every data item is sent as a 2byte integer (low byte first), and each packet framed by a hardcoded sequence. The resulting packet length is constant.

Header	Data	Footer
0x23 0x40	data0, data1, ...	0x24

Table 2: Transfer Protocol

Data can be sent with a rate of up to 200 $samples/s$.

Sending slip encoded OSC messages directly was tested, but as numerical data values in OSC are sent as 4bytes, this approximately doubles the bandwidth requirement. It would certainly be nice to define smaller OSC data types. Here, the effort to introduce the uOSC [7] variant may be a step in the right direction, although smaller data types are not yet part of the specification.

5 Calibration

MEMS [3] components such as the sensors used in this design typically reveal significant errors in offset and scale. Especially the offset can be affected considerably by the mechanical stress that occurs when mounting the component, and in some components by temperature. It is thus possible to calibrate the sensors on the finished board, and expect little change in the offset over time, but only if temperature can be kept close to the calibration conditions.

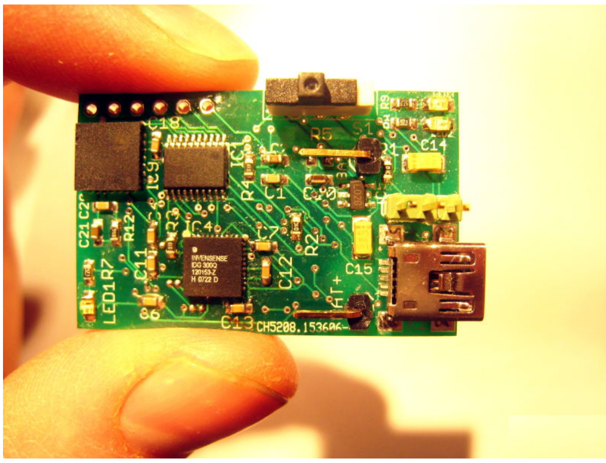

Figure 1: Front view, showing processor and gyroscopes. The bluetooth module and the accelerometer are mounted on the back.

[1] with $1g = 9.81 m/s^2$

[2] with $1g = 1gram$

[3] Micro-Electro-Mechanical System

However, in some applications even a small offset in angular velocity must be avoided. We know that for most musicians, the mean rotation speed of a device attached to the body is small, seen against the magnitude of the motions of interest. The gyroscope offset can therefore be removed almost completely by applying a highpass with a very long time constant to the output values.

Removing the offset from the accelerometer output values is more complicated, as the influence of gravity in general prevents the mean acceleration to become zero. We will later show a method to estimate the direction of gravity in respect to the sensor device, and to effectively calibrate offset and scale of both the accelerometers and the gyroscopes automatically.

6 Where is gravity?

The orientation of an object in space can be controlled very intuitively by a musician, and may therefore be a good control parameter for music applications. Also, knowing the direction of gravity will make it easy to split the measured acceleration into two parts, one part equaling gravity, and the other equaling the dynamic acceleration of the object.

To determine the orientation of the device towards gravity, the gravity vector can be estimated from the measured data. In [8], a gravity estimator that is also used to correct for offset in angular rate is constructed using a Kalman Filter. In [9], a gravity estimator that is used to compensate for the effect of gravity is constructed using rotation by the integrated angular rate.

The gravity estimator proposed here is based on continously iterated coordinate rotation, depending on the measured angular rate. It is so simple that it can be implemented in fixed point arithmetic, and without the use of trigonometric functions.

A coordinate rotation by an angle α around the X-axis can be defined by following rotation matrix:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & cos\alpha & -sin\alpha \\ 0 & sin\alpha & cos\alpha \end{pmatrix} \tag{1}$$

With a data rate of 200 samples/s, and if the assumption can be made that the typical angular rate does not exceed 360 deg/s, the expected angle differences from one sample to the next are small, and we can write $\sin\alpha \sim \alpha$ and $\cos\alpha \sim 1$, with an error of less than 0.1%. This will lead to following simplification:

$$R = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & -\alpha \\ 0 & \alpha & 1 \end{pmatrix} \tag{2}$$

The general case of a small rotation by angles α, β, γ can be written as:

$$R = \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} \tag{3}$$

If this is applied iteratively, the resulting vector will grow slowly, and in addition rounding errors can accumulate. This can be corrected by scaling the vector to unity length. Drift in the orientation of the estimated gravity can be corrected by adding a small portion $\epsilon\vec{a}$ of the measured acceleration $\vec{a}$.

The algorithm for one iteration step is as follows (for simplicity, scaling to unity length is omitted):

$$\vec{g}' = \begin{pmatrix} 1 & -\gamma & \beta \\ \gamma & 1 & -\alpha \\ -\beta & \alpha & 1 \end{pmatrix} \vec{g} + \epsilon\vec{a} \tag{4}$$

An algorithm based on this concept is implemented in the bowsense firmware.

If the additional assumption can be made that acceleration and angular rate of the object are not strongly correlated, the result of the gravity estimator could be used to calibrate both the gyroscope and the accelerometer offset automatically: The angle difference between the estimated gravity vector and the measured acceleration vector could be used to correct the scale of the gyroscopes. Each of the components x,y,z of the estimated gravity vector could be used to correct the offset of the corresponding accelerometer when small, and to correct the scale, when big.

A tested and finetuned algorithm for automatic calibration is subject to further investigation.

7 Firmware

The firmware resources are constrained by the internal 256 *byte* RAM and 8 *kbyte* FLASH memory of the C8051F530 microcontroller. In the most recent version, less than 2 *kbyte* FLASH is used for program storage, leaving space for future enhancements.

The firmware is written in C, and compiled with Keil C51. It could probably be ported to the SDCC compiler suite with little effort.

8 Conclusions

A compact 6 degrees IMU system has been presented, that can be used as a wireless controller for realtime music applications. The hardware is easily extendible, either by connecting sensors to the existing board, or by extending the design based on existing material.

A simple algorithm that calculates an estimate for the gravity vector has been shown, and the algorithm may be used for automatic sensor calibration.

More information can be found at: http://notam02.no/proj/bowsense

9 Acknowledgements

Special appreciation goes to the Norwegian Academy of Music, namely Knut Guettler, Victoria Johnson and Haakon Thelin for inspiring cooperation in the sensorbow project. Thanks also to Dan Overholt, Anders Vinjar and Giso Grimm for participating in the implementation and debugging of the transfer protocol and conversion to OSC.

References

[1] Knut Guettler, Hans Wilmers, and Victoria Johnson. Victoria counts - a case study with electronic violin bow. *Proc. ICMC2008, International Computer Music Conference, Belfast*, 2008.

[2] A. Askenfelt. Measurement of bow motion and bow force in violin playing. *Journal of the Acoustical Society of America*, 80(4):1007–1015, 1986.

[3] F. Bevilacqua, N. Rasamimanana, and E. Flety. The augmented violin project: research, composition and performance report. *Proc. NIME2006, New Interfaces for Musical Expression, Paris*, 2006.

[4] Keith A. McMillen. Stage-worthy sensor bows for stringed instruments. *Proc. NIME2008, New Interfaces for Musical Expression, Genova*, 2008.

[5] Diana Young. The hyperbow: A presision violin interface. *Proc. ICMC2002, International Computer Music Conference, Göteborg*, 2002.

[6] Dan Overholt. Musical interaction design with the create usb interface. *Proc. ICMC2006, International Computer Music Conference, New Orleans*, 2006.

[7] Andy Schmeder and Adrian Freed. uosc: The open sound control reference platform for embedded devices. *Proc. NIME2008, New Interfaces for Musical Expression, Genova*, 2008.

[8] H. J. Luinge and P. H. Veltink. Measuring orientation of human body segments using miniature gyroscopes and accelerometers. *Medical and Biological Engineering and Computing*, 43:273–282, 2005.

[9] Manuela Bandala and Malcolm Joyce. Wireless inertial sensor for tumour motion tracking. *Journal of Physics: Conference Series*, 76, 2007.